Nonuniform Sampling Coding in Networked Controlled Linear Systems

CARLOS CANUDAS-DE-WIT, KELIT CRISANTO VEGA AND JONATHAN JAGLIN
Laboratoire d'Automatique de Grenoble,
NeCS INRIA-CNRS project-teem, FRANCE.
Email: carlos.canudas-de-wit@inpg.fr.

***Abstract*— This notes concerns the non-uniform entropy (NUE) coding, and describes some of the main properties due to such a type of code.**

***Index Terms*— Entropy coding, Networked controlled systems, NCS.**

I. INTRODUCTION

This paper deals with systems interconnected by a communication network where information is transmitted via a particular coding algorithm. Many of such type of control architectures has been studied recently. Some examples are: [7], [3], [9], [10], [13], [8], [5], [1], [2], and [4] among others.

In particular, delta modulation (Δ-M) is a well-known differential coding technique used for reducing the data rate required for voice communication, see [12]. The standard technique is based on synchronizing a state predictor on emitter and receiver and just sending a onebit error signal corresponding to the innovation of the sampled data with respect to the predictor. The prediction is then updated by adding a positive or negative quantity (determined by the bit that has been transmitted) of absolute value Δ, a known parameter shared between emitter and receiver.

We have recently investigate the closed-loop properties of the Δ-M algorithm when used in the feedback loop. Our results in [2], have suggested some modification of the original form of the Δ-M algorithm to improve the closed-loop properties when used in feedback within the context of Networked controlled systems (NCS). The results showed that the stability domain and the resulting precision of the Δ-M is limited by the position of the largest unstable pole of the system. Although this can be improved by increasing the sampling rate, or by the use of extra bits [6], this possibility is clearly limited by the maximum permissible data transmission rate. Further studies, have also shown that it is possible to make the modulation gain adaptive so as to improve the global stability results [4].

The aim of this work is to explore yet another variant of the Δ-M structure to improve data transmission efficiency in the context of NCS. High compression rates can only be reached by the use of entropy coding. Entropy coding introduce redundancy and assigns some probability distribution to the events. In that way, the mean code length can be reduced.

A pre-requisite for the entropy coding strategy is to design a mechanism with the ability to quantify and to differentiate stand-still signal events, to changes in the source (label crossing detector). For instance, this can be done by defining an alphabet where the source signal information is contained in the time interval between level crossing and in the direction of the level crossing. As mentioned in [11], by assigning strings of the 2-tuple 00 to represent the time between signal level crossing, and 01 and 10 to denote the direction of level crossing, the output of the nonuniform sampling encoder contains a high probability of the 0 symbol with makes it suitable for an entropy encoder to attain a "good" overall compression ratio. A fundamental difference with the classical Δ-M algorithm is that the error is coded on the basis of a 3-valued alphabet rather than a 2-valued one. These features point toward the possibility of efficient encoding strategies.

The overall coding strategy studied here is composed of two main blocks:

1) a non uniform sampling encoder, NSE, (i.e. *asynchronous* delta modulation), including a model-based predictor, MBP, similar to the one proposed in [2], and
2) a variable length-block encoding scheme, VLE,(i.e. Run-length distortion-less encoding strategy) in which buffering is only requited at the input only.

The overall scheme is shown in Figure 1.

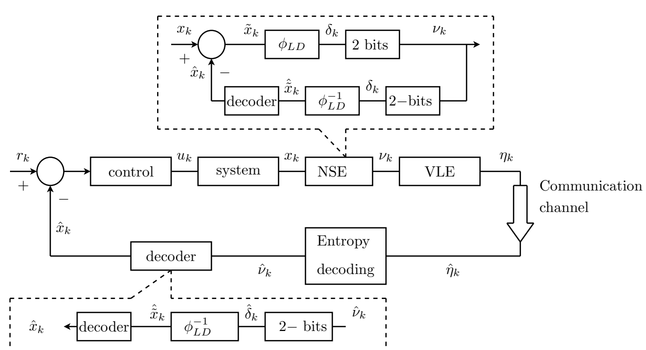

Fig. 1. Block diagram of the non uniform entropy coding in the feedback loop.

The paper aims at studding the closed-loop properties of such arrangement. Due to the fact that the VLE block is a distortion less coding, and for simplicity reasons, the focus of this study is directed toward the study of the stability properties of the first coding block only.

A. Definitions

r_k: reference signal,
x_k: system state,
$\hat{x}_k$: estimated state,
$\tilde{x}_k$: true estimated error, $\tilde{x}_k = x_k - \hat{x}_k$,
$\hat{\tilde{x}}_k$: approximated estimated error, obtained after reconstruction, i.e. $\hat{\tilde{x}}_k = \{\varphi_{LD}^{-1} \circ \varphi_{LD}\}(\tilde{x}_k)$
Δ: step interval used for level detection and to reconstruct $\hat{\tilde{x}}_k$,
δ_k: 3-level valued integer signal: $\{-1, 0, 1\}$
ν_k: 2-bits binary signal; $\{00, 01, 10\}$
η_k: variable length binary signal to be send by the channel (output of the RLE block.
u_k: control input

B. Assumptions

The hypothesis used in the results presented in this paper, are the following:

- The transmitted information is binary
- Only encoder-to-decoder information transmission is allowed (feedback between decoder to encoder is forbidden),
- Reliable noiseless channel transmission is considered (no data lost, or information distortion is considered),
- Transmission delays are neglected,

II. Problem set up

We consider the following SISO discrete-time linear system (possible unstable), of the form,

$$x_k = \frac{B(q^{-1})}{A(q^{-1})} u_k \tag{1}$$

together with a RST controller,

$$u_k = \frac{R(q^{-1})}{S(q^{-1})} \left\{ \frac{\gamma}{T(q^{-1})} r_k - \hat{x}_k \right\} \tag{2}$$

where r_k is the reference, $\hat{x}_k$ is the estimated of the system output x_k, and $R(q^{-1}), S(q^{-1}), T(q^{-1})$ are the control polynomials in the delay operator q^{-1}. They also satisfy:

$$T = RB, \quad SA + RB = A_{cl}, \quad \gamma \overset{\triangle}{=} A_{cl}(1)$$

with A_{cl} being the closed-loop polynomial, and γ the static gain needed to reach unitary zero-frequency gain. For simplicity, we will omit the use of the argument (q^{-1}) when needed.

The coding process consists in: 1) encoding the system output x_k, 2) transmitting the coding sequence through the communication channel, and 3) decoding the received information to produce the estimated $\hat{x}_k$. The complete sequence can be seen as estimation process.

When $\hat{x}_k \equiv x_k$, the above controller give the following closed-loop relation,

$$x_k = \frac{\gamma}{A_{cl}(q^{-1})} r_k$$

else ($\hat{x}_k \neq x_k$), we have,

$$x_k = \frac{\gamma}{A_{cl}(q^{-1})} r_k + W(q^{-1}) \tilde{x}_k$$

where $\tilde{x}_k = x_k - \hat{x}_k$ is the estimation error, and $W = BR/A_{cl}$. As A_{cl} defines a stable polynomial, the output x_k is keep bounded as long as $\tilde{x}_k$ is bounded as well.

The problem is then to design the coding process that defines the output $\hat{x}_k$ preserving closed-loop properties. This process is described next.

III. Coding Process

The coding (encoding/decoding) process is composed of several steps, described by the following operations:

$$x_k \underbrace{\mapsto}_{NU-enc.} \nu_k \underbrace{\mapsto}_{E-enc.} \eta_k \underbrace{\Rightarrow}_{channel} \hat{\eta}_k \underbrace{\mapsto}_{E-dec.} \hat{\nu}_k \underbrace{\mapsto}_{NU-dec.} \hat{x}_k$$

As shown in Figure 1, the encoder (respectively the inverse decoder) operation is composed of two separate blocks:

- The non uniform sampler encoder (NUE) together with a model-based predictor (MBP), this operation maps $x_k \mapsto \nu_k$ (respectively the decoder maps $\hat{\nu}_k \mapsto \hat{x}_k$), and
- A variable length encoder (VLE) mapping the 3-valued signal, ν_k to the string binary signal η_k (respectively, the decoder maps $\hat{\eta}_k \mapsto \hat{\nu}_k$).

Each of these components are described next.

A. Description of the NUE and the MBP

Elements composing the Non-Uniform Entropy coding and the model-based predictor are the following:

1) The Level Detector: ($\varphi_{LD} : \tilde{x}_k \mapsto \nu_k$), take the error signal and codes the output signals into a 3-valued one $\delta_k \in \{-1, 0, 1\}$ (or equivalent to the binary signal ν_k). That is:

$$\delta_k = \begin{cases} 1 & if \quad \text{one level is crossed uppwards} \\ 0 & if \quad \text{signal stay at the acual level} \\ -1 & if \quad \text{one level is crossed dawnwards} \end{cases}$$

Equations behind this are:

$$\begin{aligned} l_k &= \begin{cases} \lfloor \frac{\tilde{x}_k}{\Delta} \rfloor & \text{if,} \quad \tilde{x}_k > 0 \\ 0 & \text{if,} \quad \tilde{x}_k = 0 \\ \lceil \frac{\tilde{x}_k}{\Delta} \rceil & \text{if,} \quad \tilde{x}_k < 0 \end{cases} \\ \delta_k &= f(\tilde{x}_k) = \begin{cases} 0 & \text{if,} \quad l_k = l_{k-1} \\ \text{sign}\,(l_k - l_{k-1}) & \text{else} \end{cases} \end{aligned}$$

the level threshold is Δ, and the floor operator $\lfloor \cdot \rfloor$ rounds to the smaller integer, and the ceil operator $\lceil \cdot \rceil$ rounds to the larger integer.

Finally, the 3-valued signal δ_k is transform into a a 2-bits binary number $\nu_k \in \{00, 01, 10\}$, by the following operation.

$$\nu_k = \begin{cases} 00 & if \quad \delta_k = 0 \\ 01 & if \quad \delta_k = -1 \\ 10 & if \quad \delta_k = 1 \end{cases}$$

The combination '$11'$ is not used in this process. Note that δ_k is just a dummy internal variable useful to simplify the formalization of the error equations in stability studies to be presented latter, which can be formalized using real numbers.

Operation principle of the LD. The operation principe of the level detector is shown in Figure 2. The signal detection levels are uniformly spaced by the quantum Δ. The level

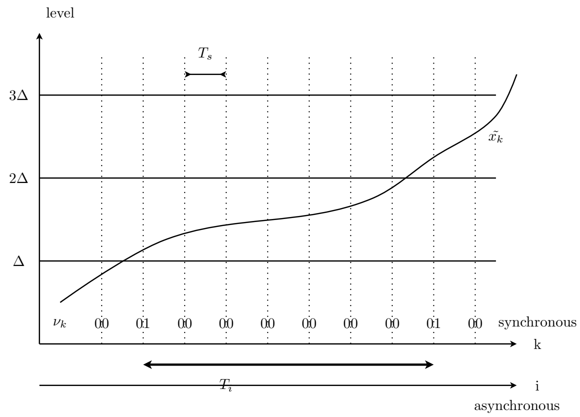

Fig. 2. Illustration of the level detector working operation principe.

detector device produces a signal (identified by $'01'$ or $'10'$) whenever a level crossing takes place. As level changes are not uniformly spaced in time, this device normally works on the basis of non-uniform sampling. However a complete asynchronous problem formulation, will also require to describe the system to be controlled in this time-asynchronous framework. Instead, and following [11], we consider quantizing the time between level crossing and encoding the quantizing time interval in uniform steps. By doing this the information transmission in synchronous.

While two symbols are used to characterize the level changes, one more symbol can be used to quantize time intervals. Then 01 indicates upward crossing, 10 downward crossing, and 00 is used to code the time-interval between crossing. To illustrate this consider the example of Figure 2. We assume that uniform samples are taken every time T_s, then m samples are taken in the time interval $T_i = t_i - t_{i-1}$ before a cross level takes place. As two level (upward)crossing happen within this interval, the information the binary representation of this situation by the level crossing detector produce the following signal,

$$01, \underbrace{00, 00, \ldots 00}_{m-pairs}, 01$$

This sequence has then high probability of $0'$s, and thus suited for entropy coding.

2) The model-based predictor: Has the role to recover the encoded signal x_k for the 3-valued binary signal ν_k. It is composed of:

- *The inverse of the level detector:* $\varphi_{LD}^{-1} : \nu_k \mapsto \delta_k \mapsto \hat{\tilde{x}}_k$, which equations are:

$$\delta_k = \begin{cases} 0 & if \quad \nu_k = 00 \\ -1 & if \quad \nu_k = 01 \\ 1 & if \quad \nu_k = 10 \end{cases}$$

  and,

$$\hat{\tilde{x}}_k = \hat{\tilde{x}}_{k-1} + \Delta \cdot \delta_k$$

- *The predictor.* Which is a model-based predictor. As it name indicated it used the target closed-loop model as a basis for it design. This structure is inspired by our previous works in [2], [4], and also in [5]. The predictor is a dynamic linear discrete-time operator that maps the output of the inverse level detector, to the signal prediction $\hat{x}_k$. Its structure depends upon the particular control used (state feedback or output feedback). For instance, for the RST-control discussed here, it has the following form:

$$\hat{x}_k = W\left[\frac{\gamma}{T} r_k + \hat{\tilde{x}}_k\right], \quad W \triangleq \frac{BR}{A_{cl}} \tag{3}$$

  Which results in the following error equation:

$$\tilde{x}_k = W\left[\tilde{x}_k - \hat{\tilde{x}}_k\right] \tag{4}$$

run sequence ν_k	output $\eta_k(n=2)$	output $\eta_k(n=3)$
01	00	000
10	01	001
0001	10	010
0010	11	011
000001	-	100
000010	-	101
000000	-	110
unused	-	111

TABLE I
RUN-LENGTH ENCODING.

B. Description of the entropy coding VLE

As mentioned in the introduction, high compression rates can only be reached by the use of entropy coding. Entropy coding introduce redundancy and assigns some probability distribution to the events. In that way, the mean code length can be optimized. Run-length codes[1], are a class of variable-length codes that are sub-optimal (when compared to the Huffman code), but have the advantage of avoiding buffering at the decoder side, and therefore reducing data transmission latency. An example used in [11] is described below.

Let $\varphi_{RL} : \nu_k \mapsto \eta_k$, describe the VLE map. Then, when a run-length code of length n is applied, the result is the output η_k which a variable-length binary signal. An example of a coding scheme of block length $n = 2$, and $n = 3$ is shown below.

Assuming that the coding sequence is independent and identically distributed, and that the upward crossing frequency equals the downwards crossing frequency, i.e.

$$p = P(00), \quad P(01) = P(10) = \frac{1}{2}(1-p)$$

then, according to [11], the mean coding length, C_L, of this scheme is:

$$C_L = 2\frac{1 - p^{(2^{(n-1)}-1)}}{1-p} bits$$

and the compaction ratio,C_R, is:

$$C_R = \frac{C_L}{n} = \frac{2}{n}\frac{1 - p^{(2^{(n-1)}-1)}}{1-p}$$

which has the limiting value: $\lim_{p\to 1} C_R = \frac{2^n - 2}{n}$.

[1]Class of coding strategy that can decode information instantaneously.

IV. Error system

Following the assumptions maid in this paper (lossless channel transmission), we then have that $\nu_k = \hat{\nu}_k$, and that $\delta_k = \hat{\delta}_k$. In this case binary variables are not needed, and hence error equation can be described by real variables only.

Introducing the following error definitions:

- $e_k = x_k - \frac{\gamma}{A_{cl}} r_k$: the tracking error,
- $\tilde{x}_k = x_k - \hat{x}_k$: the prediction error, and
- $\varepsilon_k = \tilde{x}_k - \hat{\tilde{x}}_k$: the LD error.

we have the closed-loop error system:

$$e_k = W(q^{-1})\tilde{x}_k \quad (5)$$
$$\tilde{x}_k = W(q^{-1})\varepsilon_k \quad (6)$$

with $W = BR/A_{cl}$ being the stable operator defined previously. Note that the $\varepsilon_k = \varepsilon_k(\tilde{x}_k)$, and thereby the above error equation can be seen as two systems in cascade, i.e. the output of the autonomous system (6) is the input of the stable system (5). For stability purposes it is thus sufficient to demonstrate the stability properties of the sub-system (6).

A. Properties of the LD and it inverse

Note that ε_k writes as:

$$\begin{aligned} \varepsilon_k &= \tilde{x}_k - \hat{\tilde{x}}_k \\ &= \tilde{x}_k - \varphi_{RL} \circ \varphi_{RL}^{-1}\{\tilde{x}_k\} \\ &= \tilde{x}_k - \tilde{\varphi}_{RL}\{\tilde{x}_k\} \end{aligned}$$

where $\tilde{\varphi}_{RL} \stackrel{\triangle}{=} \varphi_{RL} \circ \varphi_{RL}^{-1} : \tilde{x}_k \mapsto \hat{\tilde{x}}_k$. Note that this map is dynamic, defined by the following relation:

$$\hat{\tilde{x}}_k = \hat{\tilde{x}}_{k-1} + \Delta \cdot \delta_k \quad (7)$$

with $\delta_k = f(\tilde{x}_k)$ as defined before. System (6)-(7) can be then seen as a feedback system, as shown by Figure 3, i.e.

$$\begin{aligned} \tilde{x}_k &= W(q^{-1})\varepsilon_k \\ &= W(q^{-1})\left(\tilde{x}_k - \hat{\tilde{x}}_k\right) \\ &= W(q^{-1})\left(\tilde{x}_k - \frac{\Delta}{1-q^{-1}} f(\tilde{x}_k)\right) \end{aligned}$$

Ideally we would like that the map $\tilde{\varphi}_{RL}$ be a linear map with

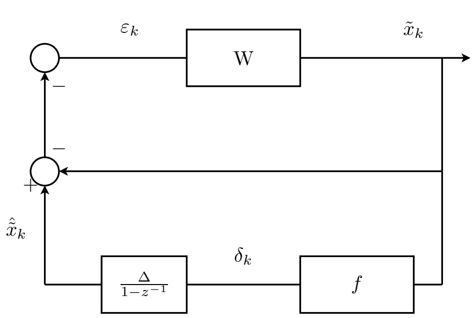

Fig. 3. Closed-loop interconnection of System (6)-(7). Jonathan: merci de faire cete figure.

unitary gain. This ideal goal is hampered by several factors:

- unknown initial conditions of $\tilde{x}_0$,
- badly chosen T_s, and Δ, and
- chattering in the neighborhood of the quantum Δ.

In particular, large sampling times T_s, and too small quantum Δ may results in signal variation of more than one level, which may leads to unrecovered bias in the estimated, leading to potential instabilities for unstable open-loop systems.

The qualitative valuation of this algorithm is shown next trough a simulation examples, and some qualitative discussions.

V. Qualitative evaluation

Consider the following simple system

$$\frac{A(q^{-1})}{B(q^{-1})} = \frac{bq^{-1}}{1 - aq^{-1}} \quad (8)$$

The controller is:

$$R = r_0 + r_1 q^{-1}, \quad S = 1 - q^{-1}, \quad T = \frac{A_{cl}(1)}{B(1)} A_{cl} = (1 - a_{cl})^2$$

obtained from the closed-loop specification given by $A_{cl} = (1 - a_{cl})^2$, and $r_0 = \frac{a+1-2a_{cl}}{b}$, $r_1 = -\frac{a-a_{cl}^2}{b}$. Note that this controller introduced a integral action as well. Parameter used in simulations are: $a = 1.1$, $b = 1$, $a_c = 0.96$, $T_s = 0.05$ (sec), $\Delta = 0.02$, and $x_0 = 0.1$.

A. Initial conditions

Initial conditions of the predictor $\hat{x}_0$ need to be synchronized with initial condition of the system, i.e. $\hat{x_0} = x_0$, in particular at the decoder side. This requires a specific initialization procedure that send this initial information before the coding algorithm is triggered. Figure 4 shows the algorithm behavior when this synchronization is performed correctly.

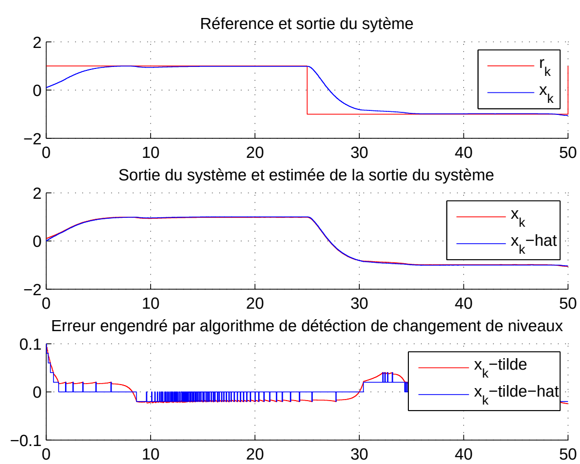

Fig. 4. Simulations with synchronized initial conditions; $x_0 = \hat{x}_0 = 0.1$. reference and output (top), output and it estimated (middle), and error variables $\tilde{x}$(red), and $\hat{\tilde{x}}$(blue)

When this synchronization is not done correctly, Figure 5 shows a bias in the estimates, which produces an error in the tracking signal. Note however, that this ill-synchronization does not causes instabilities, but just a tracking performance degradation, proportional to the initial mismatch between initial condition in the decoder estimates.

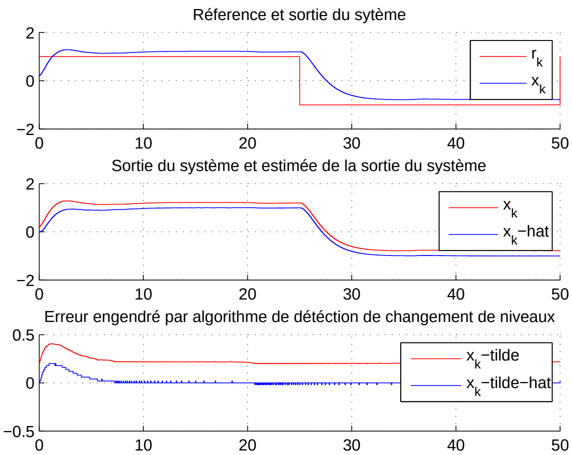

Fig. 5. Simulations with ill-synchronized initial conditions; $x_0 \neq \hat{x}_0 = 0.1$. reference and output (top), output and it estimated (middle), and error variables $\tilde{x}$(red), and $\hat{\tilde{x}}$(blue)

B. *Quantum Δ and sampling time T_s*

The value of Δ has an important impact on the quality of the estimates and the system stability. Larger values of Δ are suited for enhance stability, but it degrades the estimated quality. Inversely, small values for Δ will results in better estimated quality, but it jeopardize the system stability. Evidently, also the selection of the sampling period T_s play also an important role, and more specifically, the ratio Δ/T_s, as it will be discussed below.

Note that the coding scheme is based on the hypothesis that information is coded by a 3-valued signal δ_k, or equivalently that level changes are restricted to a maximum of one level of magnitude Δ. This can bee seen from the inverse of the level detector equation, that aims at reconstructing the estimation error trough the relation $\hat{\tilde{x}}_k = \hat{\tilde{x}}_{k-1} + \Delta \cdot \delta_k$. When the this equation is correctly initialized ($\hat{\tilde{x}}_0 = \tilde{x}_0$) in particular at the decoder side, and the change of level is limited to one, then the maximum distance between $\hat{\tilde{x}}_k$, and $\tilde{x}_0$ is given by Δ. Looking closely the error equation (5)-(6), we can see that as a consequence, the norm of the tracking error becomes proportional to Δ, i.e.

$$|e| \leqslant |W|^2 \Delta$$

the system precision is then proportional to Δ as long as $|l_k - l_{k-1}| \leqslant 1$. In order to this to happen, it is necessary that to select Δ such as the net change between two consecutive samples of $\tilde{x}$ is bounded by this value, i.e.

$$|\tilde{x}_k - \tilde{x}_{k-1}| < \Delta \qquad (9)$$

In this relation the sampling time does not appears explicitly because is embedded in the particular discrete-time notation, which have the following signification:

$$\tilde{x}_k = \tilde{x}(t_k), \quad \tilde{x}_{k-1} = \tilde{x}(t_k - T_s)$$

Thus equation (9) can also be read as:

$$|\frac{d}{dt}\tilde{x}(t)| \approx \frac{1}{T_s}|\tilde{x}(t_k) - \tilde{x}(t_k - T_s)| < \frac{\Delta}{T_s}$$

the rate of change of the estimated error signal should thus be bounded by the ration $\frac{\Delta}{T_s}$. The Figure 6, and Figure 7 shown this event. In Figure 6 $\frac{\Delta}{T_s}$ is sufficient small to keep $|l_k - l_{k-1}| \leqslant 1$, whereas in Figure 7 $|l_k - l_{k-1}| \leqslant 2$. As a consequence in this second case the estimation performance is degraded, and in certain case for even larger $\frac{\Delta}{T_s}$ (not shown here) instabilities may occur.

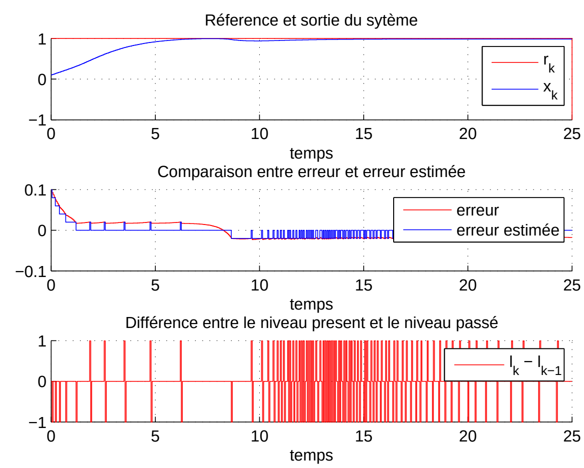

Fig. 6. Simulation results with $\Delta = 0.02$; $Te = 0.05$ yielding $|l_k - l_{k-1}| \leqslant 1$. Output and reference (upper), $\tilde{x}$ vs.$\hat{\tilde{x}}$ (middle), and $l_k - l_{k-1}$ (lower)

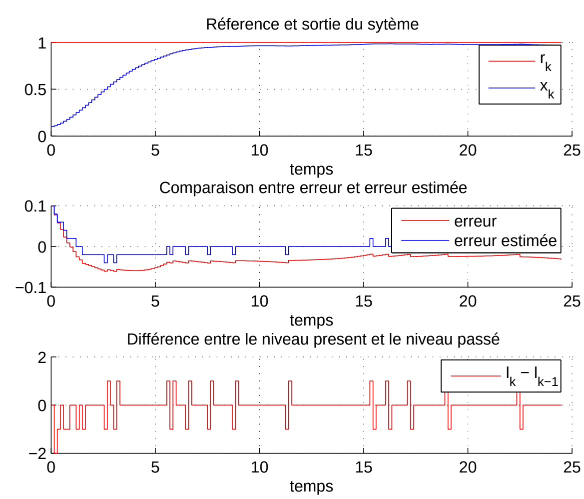

Fig. 7. Simulation results with $\Delta = 0.02$; $Te = 0.15$ yielding $|l_k - l_{k-1}| \leqslant 2$. Output and reference (upper), $\tilde{x}$ vs.$\hat{\tilde{x}}$ (middle), and $l_k - l_{k-1}$ (lower)

C. Chattering effect

Note that practically all parameters of the control scheme affects the spectrum of the evolution of δ_k, and in particular the ration Δ/T_s, but also the level of open-loop instabilities of the considered system. Figure 8 show the resulting histogram of δ_k for different simulations with different parameters. The results shows that by increasing T_s the frequency spectrum of δ_k is reduced, and hence the event $\delta_k = 0$ has higher probability to occur. We recall distributions with high roll-off will be benefic for data compression, as illustrated by the VLE algorithm. from the same figure and similar reasons, we observe that open-loop instable systems with a high degree of instability are less adapted for entropy coding. The resulting compression ratio are reported in Table II. Higher compression rates are thus obtained for the cases where p is higher, i.e. the bottom simulation in Fig.8.

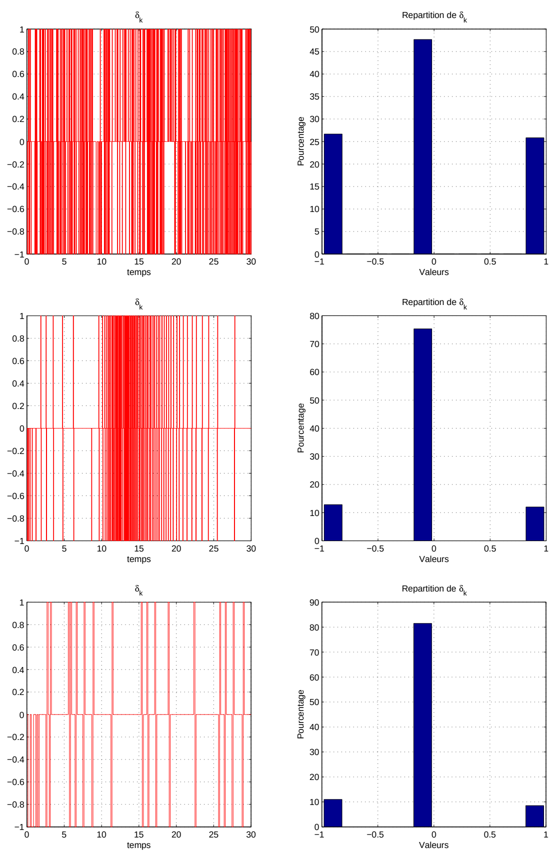

Fig. 8. Simulations with $\Delta = 0.02$;$T_e = 0.05$;$a = 1.6$ (upper), $\Delta = 0.02$; $T_e = 0.05$; $a = 1.1$ (middle), and with $\Delta = 0.02$; $Te = 0.15$; $a = 1.1$(bottom). Time evolution of δ_k (left), histogram of events (right)

case (n=3)	p	C_L	C_R
upper	.4	3.4208	1.1403
middle	.75	4.6250	1.5417
lower	.8	4.8800	1.6267

TABLE II

COMPRESSION RATIOS AND CODE LENGTH OBTAINED WITH $n = 3$ FROM DATA SHOWN IN FIG. 8.

VI. CONCLUSIONS

In this paper we have investigated the possibility to use asynchronous coding in the context of networked controlled systems. The main motivation has been to explore the benefits in terms of coding compression ratio when the NUE is combined with entropy coding strategies. In particular we have analyzed the case of run-length coding, which in spite of its sub-optimality do not require buffering at the decoder side, and hence reduce latency.

We have shown also that the predictor initial conditions, seems to be critical for the well operation of the coding algorithm. When these are not exactly known the predictor produce a bias, although stability is not happened. We have also presented some preliminary studies showing that the choice of Δ and T_s are critical for the well functioning of scheme. In particular "high" ratios of Δ/T_s are suited for stability, whereas in opposition "small" ratios of Δ/T_s are better for data compression. This underline a trade-off between stability and performance of the proposed scheme.

Finally, further analytic studies are in progress to find analytic conditions for stability, optimal selection of the ratio Δ/T_s, and eventually making this ratio time (state) varying.

REFERENCES

[1] Brockett R.-W. and Liberzon D. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45(7):1279–1289, July 2000.

[2] C. Canudas-de-Wit, F. Rubio, J. Fornes, and F. Gomez-Estern. Differential coding in networked controlled linear systems. *American Control Conference. Silver Anniversary ACC. Minneapolis, Minnesota USA*, 2006.

[3] N. Elia and S.-K. Mitter. Stabilization of linear systems with limited information. *IEEE Transaction on Automatic Control*, 46(9):1384–1400, September 2001.

[4] F. Rubio F. Gomez-Estern, C. Canudas-de-Wit and J. Fornes. Adaptive delta-modulation coding in networked controlled systems. *Submitted to American Control Conference, New York, USA USA*, 2007.

[5] Hespanha J.-P., Ortega A., and Vasudevan L. Towards the control of linear systems with minimum bit–rate. In *15th Int. Symp. Mathematical Theory of Networks and Systems (MTNS)*, Notre Dame, IL, USA, 2002.

[6] C.T. Abdallah I. Lopez and C. Canudas-de-Wit. Compensation schemes for a delta-modulation-based ncs. *Submitted to ECC'07 USA*, 2007.

[7] H. Ishii and T. Başar. Remote control of lti systems over networks with state quatization. *System and Control Letters*, (54):15–31, 2005.

[8] K. Li and J. Baillieul. Robust quatization for diginal finite communication bandwidth (dfcb) control. *IEEE Transaction on Automatic Control*, 49(9):1573–1584, September 2004.

[9] D. Liberzon. On stabilization of linear systems with limited information. *IEEE Transaction on Automatic Control*, 48(2):304–307, February 2003.

[10] Lemmon M. and Q. Ling. Control system performance under dynamic quatization: the scalar case. In *43rd IEEE Conference on Decision and Control*, pages 1884–1888, Atlantis, Paradice Island, Bahamas, 2004.

[11] J.W. Mark and T.D. Tood. A nonuniform sampling approach to data compression.

[12] J.G. Proakis. *Digital Communications*. McGraw-Hill, Inc. Series in electrical and computer enginering, 2001.

[13] S. Tan, Xi Wei, and J.-S. Baras. Numerical study of joint quatization and control under block-coding. In *43rd IEEE Conference on Decision and Control*, pages 4515–4520, Atlantis, Paradice Island, Bahamas, 2004.